UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/783,598	02/20/2004	David Wortendyke	MS1-2021US	1564

22801 7590 08/13/2007

LEE & HAYES PLLC
421 W RIVERSIDE AVENUE SUITE 500
SPOKANE, WA 99201

EXAMINER	
CHILES, AARON T	

ART UNIT	PAPER NUMBER
2109	

MAIL DATE	DELIVERY MODE
08/13/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/783,598	WORTENDYKE ET AL.	
	Examiner	Art Unit	
	Aaron Chiles	2109	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *20 February 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-43* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-43* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *20 Feb 2004* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *See Continuation Sheet*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Continuation of Attachment(s) 3). Information Disclosure Statement(s) (PTO/SB/08), Paper No(s)/Mail Date :20 Feb. 2004, 18 May 2004, 15 Nov. 2006.

DETAILED ACTION

1. Claims 1-43 have been examined.

Drawings

2. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(4) because reference character "108" has been used to designate both HW (Hardware) and Network. Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

3. The drawings are objected to because numerous reference numbers in the specification do not match labeled reference numbers in the drawings. For instance, on page 27 in the specification "removable, nonvolatile magnetic disk 626" and in the drawings "626" is labeled "operating system". Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be

labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Specification

4. The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o). Correction of the following is required: Claim 28 specifies a .NET framework. No mention of this is made in the specification. Also, Claim 43 specifies a Common Language Runtime (CLR). No mention of this is made in the specification.

In addition, paragraph one of the instant specification includes a blank space.

Claim Rejections - 35 USC § 101

5. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 10-30, 37-43 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. As per claims 10-30, 37-43, these claims recite "one or more computer-readable media". Based on the Applicant's specification, page 26, lines 16-19, that reads "Communication media typically embodies computer-readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media." However these data signals are not tangible, and cannot tangibly embody a computer program or process when embodied on the data signal. Computer program or processes are only realized within the computer when stored in a memory or storage element (such as RAM or ROM). Therefore, a data signal does not meet the "useful, concrete, and tangible" requirement as set forth in *State Street*, 149 F.3d at 1373, 47 USPQ2d at 1601-02, and hence claims 10-30, 37-43 are non-statutory under 35 U.S.C. 101.

Claim Rejections - 35 USC § 102

6. Claims 1-4, 10-13, 15 are rejected under 35 U.S.C. 102(b) as being anticipated by Aguilera et al (NPL #1, Matching Events in a Content-based Subscription System).

As per claim 1, Aguilera et al discloses a method for updating a filter engine opcode tree (Appendix A, pg 8, predicates are equivalent to opcodes), comprising the following steps:

(a) compiling a new query to derive a series of opcode objects (Sec. 3, Para. 2, "each subscription is a conjunction of *elementary predicates*);

(b) traversing the opcode tree according to the series of opcode objects until an opcode object is encountered that is not included in the opcode tree, opcode objects being represented in the opcode tree as opcode nodes (App. A, Para. 2, lines 5-7); and

(c) adding new opcode tree opcode nodes to correspond to the encountered opcode object and subsequent opcode objects in the series of opcode objects (App. A, Para. 2, lines 7-8).

As per claim 2, Aguilera et al also discloses wherein one or more of the steps are performed dynamically at runtime (pg. 2, Col. 1, lines 22-24). It is a well-known goal of subscription server systems, as disclosed by Aguilera, to maximize availability. As such, it is anticipated by Aguilera that the incremental updates would be done at run-time.

As per claim 3, Aguilera et al also discloses further comprising performing steps (b) and (c) in a component of the filter engine (code listing, pg. 9, the pre-processing steps are a procedure contained within the tree matching algorithm, and as such, do not pertain to a stand alone program, and are therefore inherently part of the tree matching algorithm (filter engine)).

As per claim 4, Aguilera et al also discloses further comprising executing the opcode tree against an input to evaluate the new query and one or more other queries against the input (Sec. 2, Par. 2, lines 1-7).

As per claim 10, Aguilera et al discloses a filter engine (Section 3, tree matching algorithm) stored on one or more computer-readable media, comprising: a filter table that includes a plurality of queries, at least two of the queries including a common prefix (Sec. 3, matching tree); a compiler configured to compile each query into a series of opcode blocks (App. A, pre-processor); an opcode tree stored in memory and including opcode nodes that each correspond to an opcode block such that executing the opcode nodes evaluates the plurality of queries (pg. 2, col 1, lines 29-35), at least one opcode node corresponding to an opcode block included in the common prefix (Appendix A); and an opcode merger configured to merge a new query to the opcode tree by adding at least one opcode node that corresponds to the new query to the opcode tree (pg. 2, col. 1, lines 22-24).

As per claim 11, Aguilera et al discloses the opcode merger further configured to traverse the opcode tree to determine if an opcode node corresponding to the new query already exists in the opcode tree and add new opcode nodes that correspond to query opcode blocks that are not already included in the opcode tree (App. A, pre-processing).

As per claim 12, Aguilera et al also discloses wherein opcode nodes corresponding to opcode blocks included in a common prefix are represented as a shared segment in the opcode tree (App. A).

As per claim 13, Aguilera et al also discloses wherein queries are merged into the opcode tree dynamically at runtime (pg. 2, Col. 1, lines 22-24). It is a well-known goal of subscription server systems, as disclosed by Aguilera, to maximize availability. As such, it is anticipated by Aguilera that the incremental updates would be done at run-time.

As per claim 15, Aguilera et al discloses the compiler being further configured to create opcode objects that are configured to merge themselves into an appropriate location in the opcode tree (App. A, pre-processing algorithm).

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 5,14, 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Aguilera et al (NPL #1, Matching Events in a Content-based Subscription

System), in view of Java API (NPL #3, excerpt from Java 2 Platform Std. Ed. V1.4.2).

As per claim 5 although Aguilera et al does not explicitly disclose the method as recited in claim 1, further comprising: receiving a request to remove a first query from the opcode tree; identifying one or more opcode nodes in the opcode tree that correspond to the first query; removing any identified opcode node that does not correspond to a second query. Aguilera et al does state that the disclosed algorithm has space complexity that is linear with respect to the number of subscriptions (Sec. 4, heading "Space Complexity"). As such, it must inherently include a method of removing expired queries. As Aguilera is silent on the point of subscription (query) removal, one of ordinary skill in the art would know to execute the steps described in claim 5, in order to remove expired queries, and maintain the minimum size of the tree. This is further shown in the Java API. The Node Interface has methods to facilitate removal of nodes.

As per claim 14, although Aguilera et al does not explicitly disclose further comprising Xpath queries in the plurality of queries, it would be obvious to one of ordinary skill in the art to use this system in conjunction with Xpath queries. Evidence for this can be seen by the fact that Aguilera has been cited multiple times by papers dealing with XML and Xpath, and listed as related research. The systems are not incompatible, as Aguilera's tests could be tests against Xpath attributes.

Claim 31 is substantially the same as claim 5, and is rejected on the same grounds.

9. Claims 6-9, 16-18, 32-36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Aguilera et al as applied to claim 1 above, and further in view of Graefe et al (NPL #2, Dynamic Query Evaluation Plans).

As per claims 6-8, Aguilera et al discloses the limitations of claim 1, as above. In addition, Aguilera discloses an indexed look-up routine (end of section 5). However, Aguilera et al does not disclose comprising updating a branch node in the opcode tree.

However, Graefe et al discloses comprising updating a branch node in the opcode tree to add a reference to the new opcode nodes, the branch node being referenced from a parent opcode node that corresponds to a last opcode object from the series of opcode objects that was found in the traversal of the opcode tree (pg. 361, last paragraph of section 5, "If there is no gain in using a dynamic access module, the decision tree can by an empty function"), the branch node further comprising updating the branch node to include an indexed lookup routine that references several dependent opcode nodes that perform a similar function, further comprising analyzing opcode nodes that depend from the branch node and including the indexed lookup routine only if including the indexed lookup routine provides more efficient processing of the dependent nodes that a generic branch node processing routine (Section 5, dynamic query evaluation plans include a decision procedure which is used at initialization time, and intermittently thereafter to choose the optimal evaluation strategy).

Additionally, on pg. 362, in the second full paragraph, Graefe et al disclose that the choose plan operator can be inserted at any appropriate location.

It would have been obvious to one of ordinary skill in the art to evaluation system of Aguilera et al with the choose plan operators of Graefe et al in order to increase it's efficiency as described in Graefe et al. In addition, the fact that Graefe et al pertains to normal query evaluations is immaterial to it's utility in combination to Aguilera, as Graefe does still deal with complex queries, and is still in the art of query evaluation.

As per claim 9 Aguilera et al does not explicitly disclose the indexed lookup routine further comprising one of the following routines: a hash routine; a routine that uses tries; an interval tree routine. However, as Aguilera is silent as to which indexing method to use, it would obvious to one of ordinary skill to choose an efficient indexing method based on the attribute in question, which includes a hash routine; a routine that [or], an interval tree routine. For instance, in Graefe on page 360 "using a hash index"

As per claim 16, Aguilera discloses the limitations of claim 10, as above. However, Aguilera does not disclose the opcode merger being further configured to perform the following steps: when an opcode node will depend from a branch node when added to the opcode tree, identifying one or more child opcode nodes that depend from the branch opcode; and implementing an optimized branch node that includes an optimized indexed lookup procedure if such implementation would increase branch processing efficiency and referencing the opcode node from the optimized branch node.

However, Graefe et al teaches the opcode merger being further configured to perform the following steps: when an opcode node will depend from a branch node when added to the opcode tree, identifying one or more child opcode nodes that depend from the branch opcode; and implementing an optimized branch node that includes an optimized indexed lookup procedure if such implementation would increase branch processing efficiency and referencing the opcode node from the optimized branch node (pg. 361, last paragraph of section 5, "If there is no gain in using a dynamic access module, the decision tree can by an empty function").

As explained in claims 7-9 above, Graefe teaches a dynamic optimization technique that can be placed at any point in a query evaluation plan, such as a branch node input to hold the optimization algorithm. Further, it would be obvious to one of ordinary skill in the art to place the re-optimization calculations from Graefe into the pre-processing algorithm (opcode merger) of Aguilera in order to prevent redundant recalculation of these optimization plans, leaving the optimized techniques in the query evaluation plan (at the branch nodes). This combination provides Aguilera's system with more efficient calculation techniques, and Graefe's system with less redundant recalculation of optimization techniques.

Claim 17 is substantially the same as claim 9, and is rejected on the same grounds.

As per claim 18, Aguilera et al and Graefe et al teach the limitations of claim 10, as above. In addition, this system inherently has the property that the opcode merger is further configured to restore an optimized branch node to a generic branch node when the optimized branch node is no longer more efficient that the generic branch node (last paragraph of section 5, lines 4-6). The choose-plan is evaluated each time, and causes the most efficient query evaluation technique to be used. If this is the usual sequential method, then it does so.

It would be obvious to one of ordinary skill in the art to place the re-optimization calculations from Graefe into the pre-processing algorithm (opcode merger) of Aguilera in order to prevent redundant recalculation of these optimization plans, leaving the optimized techniques in the query evaluation plan (at the branch nodes). This combination provides Aguilera's system with more efficient calculation techniques, and Graefe's system with less redundant recalculation of optimization techniques.

As per claim 32, the limitations of claim 31 are rejected as above. In addition, Aguilera et al inherently teaches the step of modifying a branch node that references an opcode node that is removed from the opcode tree. This is considered inherent in the system of claim 31, as it would fail if it did not provide this step, with the minimal requirement of removing the reference to the removed node.

As per claim 33, the limitations of claim 32 are rejected as above. In addition the modifying further comprises removing the branch node if the branch node references only one other opcode node other than the opcode node to be removed is considered obvious because having a branch node where the tree no longer branches is superfluous and would waste unneeded space. As such, it would be obvious to one of ordinary skill to remove these unnecessary nodes.

As per claim 34-35, the limitations of claim 32 are rejected as above. In addition, it would be obvious to one of ordinary skill in the art to evaluate the choose-plan node, as explained for numerous claims above (7-9, 18, etc), during the removal of an opcode node, in order to continue to provide it's intended features.

Claim 36 is substantially the same as claim 9, and is rejected on the same grounds.

10. Claims 19-30, 37-43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Aguilera et al (NPL #1, Matching Events in a Content-based Subscription System), in view of Graefe et al (NPL #2, Dynamic Query Evaluation Plans), in further view of Java API (NPL #3, excerpt from Java 2 Platform Std. Ed. V1.4.2).

As per claim 19, Aguilera et al and Graefe et al teach a compiler stored on one or more computer-readable media containing computer-executable instructions for performing the following steps: receiving a query to be added to an opcode tree that represents a plurality of queries, at least two of which include similar prefixes; and compiling a query to produce one or more opcode objects, as explained for claims 1, 10, above. However, Aguilera et al and Graefe et al do not teach opcode objects that are each configured to merge into the opcode tree as an opcode node by determining an appropriate location in the tree to merge, and merging into the tree in accordance with a node context of the appropriate location.

However, as Aguilera and Graefe's techniques are language independent (the exemplary implementation's are not, however, the techniques are), and considering that XML and Xpath were standard in usage at the time of the invention, it would have been obvious to one of ordinary skill to use XML and Xpath to implement these functions.

The Java API teaches an extensible Node interface that includes functions that if extended could be used to provide opcode objects that are each configured to merge into the opcode tree as an opcode node by determining an appropriate location in the tree to merge, and merging into the tree in accordance with a node context of the appropriate location. Java Interfaces are designed to provide basic functions, and then be extended for usage. The Java API specifically states that it to be used with a Document Object Model, which is used with XML and Xpath. It would be trivial for one of ordinary skill in the art to include language to allow these Java Nodes to insert

themselves into a query tree. Numerous tree insertion, sorting, and removal algorithms are well known in the art.

It would be obvious to one of ordinary skill in the art to combine the teachings of Java with the techniques of Aguilera and Graefe because they belong to the same art of efficient query evaluation.

As per claim 20, the limitations of claim 19 are rejected as above. In addition, the combination inherently provides opcode objects that are further configured to merge into the opcode tree only if an identical opcode object corresponding to a similar query prefix is not already included in the opcode tree. This functionality is required if the nodes as provided by claim 19 are used with the matching tree of Aguilera. The system would not provide the increased efficiency intended by Aguilera if it allowed duplication of nodes.

As per claim 21, the limitations of claim 19 are rejected as above. In addition, it would have been obvious to one of ordinary skill in the art to use the Xpath language in the implementation of this system as Xpath was considered to be the standard at the time of the invention. Important factors when deciding implementation details, such as language, are interoperability and future usage. Using a language and specification that are standard increases the interoperability and lifespan of programs (See "Benefits of Standards, NPL #7).

As per claim 22-24, the limitations of claim 19 are rejected as above. In addition, the further limitations are substantially the same as those recited in claim 16-18, and are rejected on the same basis.

As per claim 25, the limitations of claim 19 are rejected as above. In addition, Aguilera, Graefe, and Java API teach the compiler is configured to receive the query and generate the opcode at runtime (pg. 2, Col. 1, lines 22-24); and the opcode node is configured to merge itself into the opcode tree at runtime (pg. 2, Col. 1, lines 22-24). It is a well-known goal of subscription server systems, as disclosed by Aguilera, to maximize availability. As such, it is anticipated by Aguilera that the incremental updates would be done at run-time. In addition, moving portions of this functionality to the opcode nodes does not make the system unable to stay active while updating.

As per claim 26, the opcode object as described would be required and inherent in a system as composed in claim 19. As such, this claim is also rejected on the same grounds.

As per claim 27, the limitations of claim 26 are rejected as above. In addition, Aguilera, Graefe, and Java API teach perform[ing] the recited steps dynamically at runtime (pg. 2, Col. 1, lines 22-24). It is a well-known goal of subscription server systems, as disclosed by Aguilera, to maximize availability. As such, it is anticipated by Aguilera that the incremental updates would be done at run-time. In addition, moving

portions of this functionality to the opcode nodes does not make the system unable to stay active while updating.

As per claim 28, the limitations of claim 26 are rejected as above. However, as the .NET framework is a proprietary specification, it is impossible for the examiner to determine if the .NET framework has the same functionality as Java v1.4.2. However, it is considered obvious to one of ordinary skill in the art to implement these functions within the .NET framework, given the teachings of the Java API. In addition, Java and .NET are considered to be alternatives to each other, both being class libraries designed to run on a virtual machine. Therefore, it is considered an obvious implementation detail to choose either Java or .NET based on the developer's familiarity.

As per claim 29, this claim is substantially the same as claim 22 above. The difference that the node makes the decision that the node performs the steps is supplied by claim 26 above. It would be obvious to one of ordinary skill in the art to include the re-optimization evaluations from Graefe into the self-inserting nodes of Java in order to preserve it's beneficial features of providing re-optimization at relevant times, without performing this evaluation unnecessarily often.

As per claim 30, the claim is rejected on the same grounds as claim 29 above. The decision to either add or remove an optimization technique is provided by Graefe, and would be decided upon at the same time.

As per claim 37, 38-39, 40, 41, 42 this claim is substantially the same as claim 10, 16-17, 21, 10, 19, respectively, above. Instructions stored on computer-readable media are the same as computer readable media containing instructions. These claims are, therefore, rejected on the same grounds as above.

As per claim 43, the limitations of claim 37 are rejected as above. In addition, the CLR is an execution environment for the .NET library, and is an alternative to the Java Virtual Machine, which Java runs on. Therefore, it would be necessary to use the CLR if one chooses to use .NET instead of Java for an implementation language.

Conclusion

11. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. NPL #4 is the W3C Recommendation that shows that XML and Xpath were standards at the time of the invention. NPL #5 (Optimization of Dynamic Query Evaluation Plans) is an extension of the work shown in NPL #2, for optimization done at different intervals. NPL #6 (Index Structures for Selective Dissemination of Information Under the Boolean Model) is general related art, and contains many features similar to the disclosed invention.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aaron Chiles whose telephone number is (571) 270-3424. The examiner can normally be reached on Monday-Friday, 7:30 a.m. - 5:00 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ramesh Patel can be reached on (571) 272-3688. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

ac

Aaron Chiles
8/7/07

RAMESH B. PATEL
SUPERVISORY PATENT EXAMINER
8/7/07